AMENDMENTS TO THE CLAIMS

Claims 1-5 (Cancelled)

6. (Currently Amended) A method for detecting and remediating a memory leak, the method comprising ~~the steps of~~:

establishing an aging value for an object instance created in memory;

establishing an alive value for the object instance, the alive value based upon whether said object instance is referenced by another object instance in the memory;

resetting said aging value when said object instance is referenced by an executing process;

incrementing said aging value during a garbage collection pass when said object instance had not been referenced by an executing process since a previous garbage collection pass; and[[,]]

processing said object instance as a loiterer [[when]] upon

said aging value exceeding ~~exceeds~~ a threshold value or

said object instance not established as being alive based upon the alive value~~,~~

~~processing said object instance as a loiterer~~.

7. (Currently Amended) The method of claim 6, wherein said establishing the aging value ~~step further~~ comprises ~~the steps of~~:

locating equivalent object instances in said memory; and[[,]]

processing said equivalent object instances in said memory as loiterers.

8. (Currently Amended) The method of claim 6, wherein said processing ~~step~~ comprises at least one of clearing at least one cache in memory, and reporting said object instance as a loiterer in a log file.

9. (Currently Amended) The method of claim 6, further comprising the step of foregoing said processing ~~step where~~ upon said object instance belonging ~~belongs~~ to a specified exempt class.

Claims 10-12 (Cancelled)

13. (Currently Amended) A machine readable storage having stored thereon a computer program for detecting and remediating a memory leak, the computer program comprising a routine set of instructions for causing a ~~the~~ machine to perform ~~the steps of~~:

establishing an aging value for an object instance created in memory;

establishing an alive value for the object instance, the alive value based upon whether said object instance is referenced by another object instance in the memory;

resetting said aging value when said object instance is referenced by an executing process;

incrementing said aging value during a garbage collection pass when said object instance had not been referenced by an executing process since a previous garbage collection pass; and[[,]]

processing said object instance as a loiterer [[when]] upon

said aging value exceeding ~~exceeds~~ a threshold value or

said object instance not established as being alive based upon the alive value~~, processing said object instance as a loiterer~~.

14. (Currently Amended) The machine readable storage of claim 13, wherein said establishing the aging value ~~step further~~ comprises ~~the steps of~~:

locating equivalent object instances in said memory; and[[,]]

processing said equivalent object instances in said memory as loiterers.

15. (Currently Amended) The machine readable storage of claim 13, wherein said processing ~~step~~ comprises at least one of clearing at least one cache in memory, and reporting said object instance as a loiterer in a log file.

16. (Currently Amended) The machine readable storage of claim 13, further comprising the step of foregoing said processing ~~step where~~ upon said object instance belonging ~~belongs~~ to a specified exempt class.

17. (New) An autonomic memory leak detection and remediation hardware system comprising:

a memory;

a processor configured for

establishing an aging value for an object instance created in memory;

establishing an alive value for the object instance, the alive value based upon whether said object instance is referenced by another object instance in the memory;

resetting said aging value when said object instance is referenced by an executing process;

incrementing said aging value during a garbage collection pass when said object instance had not been referenced by an executing process since a previous garbage collection pass; and

processing said object instance as a loiterer upon

said aging value exceeding a threshold value or

said object instance not established as being alive based upon the alive value.

18. (New) The hardware system of claim 6, wherein the one processor is further configured for:

locating equivalent object instances in said memory; and

processing said equivalent object instances in said memory as loiterers.

19. (New) The hardware system of claim 6, wherein said processing includes at least one of

clearing at least one cache in memory, and

reporting said object instance as a loiterer in a log file.

20. (New) The hardware system of claim 6, wherein the one processor is further configured for:

foregoing the processing upon said object instance belonging to a specified exempt class.